Clustering in Machine Learning

Discovering Hidden Patterns in Data

Ahmed BADI

ahmedbadi905@gmail.com

December 17, 2025

Outline

1. Introduction
2. Distance and Similarity
3. K-Means Clustering
4. Hierarchical Clustering
5. DBSCAN
6. Gaussian Mixture Models
7. Evaluation Metrics
8. Algorithm Comparison
9. Applications
10. Practical Considerations
11. Conclusion

What is Clustering?

Definition

Clustering is an *unsupervised* machine learning technique that groups similar data points together without prior labeling.

Key Insight: Like organizing books in a library by topic without instructions.

- No labeled examples required
- Discovers hidden structure in raw data
- Most real-world data is unlabeled

Applications:

- Customer segmentation
- Document organization
- Anomaly detection
- Gene clustering

Hard vs Soft Clustering

Hard Clustering

- Each point belongs to *exactly one* cluster
- Binary assignment: 0 or 1
- Example: K-Means

$$x_i \in C_j \text{ for exactly one } j$$

Soft Clustering

- Points have *probability* of membership
- Probabilistic: 0 to 1
- Example: Gaussian Mixture Models

$$\sum_{j=1}^{K} p_{ij} = 1$$

Distance Metrics

Euclidean Distance (Most Common)

$$d(\mathrm{x}, \mathrm{y}) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

Manhattan Distance

$$d(\mathrm{x}, \mathrm{y}) = \sum_{i=1}^{d} |x_i - y_i|$$

Cosine Similarity (Text Data)

$$\text{similarity}(\mathrm{x}, \mathrm{y}) = \frac{\mathrm{x}^T \mathrm{y}}{\|\mathrm{x}\| \|\mathrm{y}\|}$$

K-Means: The Algorithm

1. **Initialize:** Randomly select K points as initial centroids
2. **Assignment:** Assign each point to the nearest centroid

$$C_k = \{\mathrm{x}_i : \|\mathrm{x}_i - \boldsymbol{\mu}_k\| \text{ is minimum}\}$$

3. **Update:** Recalculate centroids

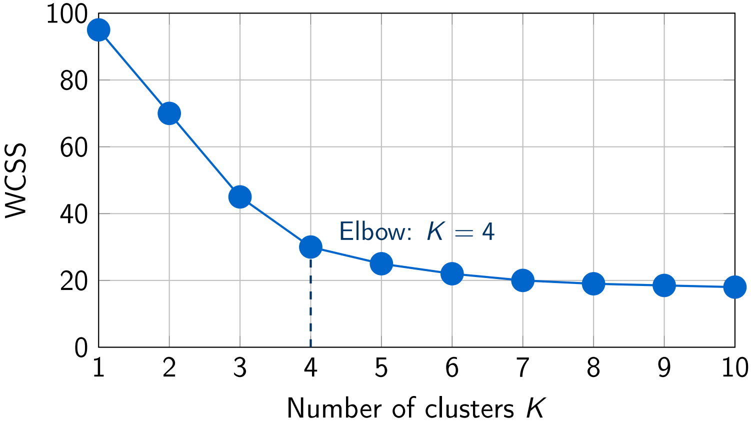
$$\boldsymbol{\mu}_k = \frac{1}{|C_k|} \sum_{\mathrm{x}_i \in C_k} \mathrm{x}_i$$

4. **Repeat:** Steps 2–3 until convergence

Objective: Minimize within-cluster sum of squares (WCSS)

$$\min \sum_{k=1}^{K} \sum_{\mathrm{x}_i \in C_k} \|\mathrm{x}_i - \boldsymbol{\mu}_k\|^2$$

Choosing K: The Elbow Method



Rule: Look for the "elbow" where WCSS levels off.

K-Means: Pros and Cons

Advantages

- Simple and intuitive
- Fast: $O(nKdi)$
- Scales to large data
- Works well with spheres

Limitations

- Must specify K
- Sensitive to initialization
- Assumes spherical clusters
- Sensitive to outliers

Hierarchical Clustering: Bottom-Up

Agglomerative Approach:

1. Start: Each point is its own cluster
2. Find two closest clusters
3. Merge them
4. Repeat until one cluster remains

Result: A dendrogram (tree) showing hierarchical structure

Advantages:

- No need to specify K beforehand
- Produces hierarchy (exploratory)
- Deterministic

Limitations:

- Expensive: $O(n^2 \log n)$ or $O(n^3)$
- Greedy (irreversible merges)

Linkage Criteria

How to measure distance between clusters?

Single Linkage : Minimum distance between clusters

$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$$

Complete Linkage : Maximum distance between clusters

$$d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$$

Average Linkage : Average distance between all pairs

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)$$

DBSCAN: Density-Based Clustering

Key Idea

Clusters are dense regions separated by sparse regions. Natural identification of noise/outliers.

Parameters:

- ϵ: Radius defining neighborhood
- MinPts: Minimum points to form dense region

Point Types:

- **Core point**: $\geq$ MinPts within ϵ
- **Border point**: Within ϵ of core but not core
- **Noise point**: Neither core nor border

DBSCAN: Pros and Cons

Advantages

- Arbitrary shapes
- Detects outliers
- No need for K
- Robust to noise

Limitations

- Sensitive to ϵ, MinPts
- Varying density issues
- High-dim challenges
- Non-deterministic

Gaussian Mixture Models (GMM)

Probabilistic Model

Data is generated from a mixture of K Gaussian distributions:

$$p(\mathrm{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathrm{x}|\boldsymbol{\mu}_k, \Sigma_k)$$

Parameters:

- π_k: Mixing weights (probabilities)
- $\boldsymbol{\mu}_k$: Mean of component k
- Σ_k: Covariance of component k

Soft clustering: Each point has probability of belonging to each component.

EM Algorithm for GMM

Expectation-Maximization:

E-step: Calculate responsibility (posterior probability)

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(\mathrm{x}_i | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathrm{x}_i | \boldsymbol{\mu}_j, \Sigma_j)}$$

M-step: Update parameters

$$N_k = \sum_{i=1}^{n} \gamma_{ik}$$

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{n} \gamma_{ik} \mathrm{x}_i$$

$$\pi_k = \frac{N_k}{n}$$

Repeat E and M steps until convergence.

GMM: Pros and Cons

Advantages

- Soft clustering
- Flexible shapes
- Statistical foundation
- Probabilistic

Limitations

- Must specify K
- Sensitive to init.
- Computationally expensive
- Gaussian assumption

Silhouette Score

Definition

For each point i:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Interpretation:

- $a(i)$: Average distance to points in same cluster
- $b(i)$: Average distance to points in nearest cluster
- Range: $[-1, 1]$
- Higher is better (≈ 1 is excellent)

Use case: Good for evaluating any clustering algorithm

Other Evaluation Metrics

Davies-Bouldin Index

$$DB = \frac{1}{K} \sum_{i=1}^{K} \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)} \right)$$

Lower is better. Measures average similarity.

Calinski-Harabasz Index

$$CH = \frac{SS_B/(K-1)}{SS_W/(n-K)}$$

Higher is better. Ratio of between/within variance.

Note: These are *internal* metrics (no ground truth needed).

Clustering Algorithms at a Glance

Algorithm	Shape	Scalability	Need K?	Noise Handle
K-Means	Spherical	Excellent	Yes	Poor
Hierarchical	Flexible	Poor	No	Poor
DBSCAN	Arbitrary	Good	No	Excellent
GMM	Ellipsoid	Moderate	Yes	Moderate

Quick Decision Guide:

- Spherical, known K: K-Means
- Hierarchy needed: Hierarchical
- Arbitrary shapes, noisy: DBSCAN
- Soft membership: GMM

Real-World Applications

- **Customer Segmentation**: Purchase behavior, demographics → marketing
- **Image Segmentation**: Pixel clustering by color/texture for computer vision
- **Document Clustering**: News/papers by topic using text features
- **Anomaly Detection**: Points not fitting clusters → fraud, intrusions
- **Recommendation Systems**: Cluster users/items to find similar groups
- **Gene Clustering**: Biology research to understand genetic relationships

Data Preprocessing

Essential Steps:

1. **Feature Scaling**: Standardize to mean 0, variance 1

$$x_i' = \frac{x_i - \mu}{\sigma}$$

   Without this, large-range features dominate distance!

2. **Dimensionality Reduction**: Use PCA/t-SNE for high-dimensional data (curse of dimensionality)

3. **Handle Missing Values**: Impute or remove before clustering

Workflow: Best Practices

1. **Explore Data**: Visualize, understand features
2. **Preprocess**: Scale, handle outliers, reduce dimensions
3. **Try Multiple Algorithms**: Start simple (K-Means), then explore others
4. **Evaluate**: Use silhouette score, Davies-Bouldin, etc.
5. **Validate with Domain Knowledge**: Does the clustering make sense?
6. **Iterate**: Refine based on results

Remember: Often no single "correct" answer—different clusterings reveal different aspects!

Key Takeaways

- **Clustering** discovers structure in unlabeled data
- **Four main approaches**:
  - K-Means: Fast for spherical clusters
  - Hierarchical: Exploratory with dendrograms
  - DBSCAN: Handles arbitrary shapes and noise
  - GMM: Probabilistic with soft assignments
- **No single best algorithm**—choose based on your data and goals
- **Preprocessing matters**: Scale features, handle high dimensions
- **Evaluation is critical**: Use multiple internal metrics

Thank You!

Questions?

ahmedbadi905@gmail.com
linkedin.com/in/badi-ahmed